



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/710,356	07/02/2004	Steven T. Shaughnessy	BORL/0222.00	4355
28653	7590	12/11/2007	EXAMINER	
JOHN A. SMART			SANDERS, AARON J	
708 BLOSSOM HILL RD., #201			ART UNIT	PAPER NUMBER
LOS GATOS, CA 95032			2168	
			MAIL DATE	DELIVERY MODE
			12/11/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/710,356	SHAUGHNESSY, STEVEN T.	
Examiner	Art Unit		
Aaron Sanders	2168		

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
 - If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
 - Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 13 August 2007.

2a) This action is **FINAL**. 2b) This action is non-final.

3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-3,5-8,10-19,21-24 and 26-30 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5) Claim(s) _____ is/are allowed.

6) Claim(s) 1-3,5-8,10-19,21-24 and 26-30 is/are rejected.

7) Claim(s) _____ is/are objected to.

8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.

10) The drawing(s) filed on 02 July 2004 is/are: a) accepted or b) objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
a) All b) Some * c) None of:
1. Certified copies of the priority documents have been received.
2. Certified copies of the priority documents have been received in Application No. _____.
3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) Notice of References Cited (PTO-892) 4) Interview Summary (PTO-413)
2) Notice of Draftsperson's Patent Drawing Review (PTO-948) Paper No(s)/Mail Date. ____ .
3) Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date . 5) Notice of Informal Patent Application
6) Other: ____ .

DETAILED ACTION

Response to Appeal Brief

In view of the appeal brief filed on 13 August 2007, PROSECUTION IS HEREBY REOPENED. The new grounds of rejection are set forth below.

To avoid abandonment of the application, Appellant must exercise one of the following two options:

- (1) file a reply under 37 CFR 1.111 (if this Office action is non-final) or a reply under 37 CFR 1.113 (if this Office action is final); or,
- (2) initiate a new appeal by filing a notice of appeal under 37 CFR 41.31 followed by an appeal brief under 37 CFR 41.37. The previously paid notice of appeal fee and appeal brief fee can be applied to the new appeal. If, however, the appeal fees set forth in 37 CFR 41.20 have been increased since they were previously paid, then appellant must pay the difference between the increased fees and the amount previously paid.

A Supervisory Patent Examiner (SPE) has approved of reopening prosecution by signing below:



CHRISTIAN CHACE
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Specification

The use of numerous trademarks, including BTRIEVE, JAVA, ORACLE, MACINTOSH, LINUX, SOLARIS, FREEBSD, and UNIX, has been noted in this application. They should be capitalized wherever they appear and be accompanied by the generic terminology. Although the use of trademarks is permissible in patent applications, the proprietary nature of the marks should be respected and every effort made to prevent their use in any manner that might adversely affect their validity as trademarks.

Drawings

The drawings are objected to under 37 CFR 1.83(a). The drawings must show every feature of the invention specified in the claims. Therefore, the method of claims 1-3, 5-8, and 10-16 must be shown or the features canceled from the claims. No new matter should be entered.

Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as "amended." If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency.

Additional replacement sheets may be necessary to show the renumbering of the

Art Unit: 2168

remaining figures. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either “Replacement Sheet” or “New Sheet” pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Claim Objections

As per claim 1, the term “preserving” is unclear. It is not clear from the specification how the logical undo operation is preserved.

Claims 1 and 17 provide for the use of a shared cache, i.e. “for use by multiple databases”; the use of a shadow cache, i.e. for storing database blocks that overflow the cache view; and performing the logical undo operation, i.e. “in order to reconstruct a transactionally consistent prior version of the given database.” Since the claim does not set forth any steps involved in these processes, it is unclear what Applicant is intending to encompass. While a recitation of use is not improper *per se*, it is not given patentable weight when interpreting the claims.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 1-3, 5-8, 10-19, 21-24, and 26-30 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

The disclosed subject matter lacks a practical application of a judicial exception (law of nature, abstract idea, naturally occurring article/phenomena) since it fails to produce a tangible result.

Specifically, the disclosed subject matter does not produce a tangible result because it fails to produce a result that is limited to having real world value rather than a result that may be interpreted to be abstract in nature as, for example, a thought, a computation, or manipulation of data. More specifically, the disclosed subject matter provides for creating a view of the database at a particular point in time for performing read-only transactions. This produced result remains in the abstract because there is no result sent to another system or output displayed to a user. Thus, the disclosed result fails to achieve the required status of having real world value.

As per claims 17-19, 21-24, and 26-30, the instant claims are directed to software *per se*. Independent claim 17 recites a computer program *per se* and functional descriptive material consisting of data structures and computer programs, which impart functionality when employed as a computer component. As such, the instant claims are not limited to statutory subject matter and are therefore non-statutory. See Diehr, 450 U.S. at 185-86, 209 USPQ at 8.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1-3, 5-8, 10, 12, 13, 15-19, 21-24, 26, 28, and 29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hayashi et al., U.S. 5,715,447 (Hayashi), in view of Loaiza et al., U.S. 6,618,822 (Loaiza) and Luo et al., U.S. 6,990,503 (Luo).

1. In a database system employing a transaction log, an improved method for restoring databases to a consistent version, the method comprising:

providing a shared cache storing database blocks for use by multiple databases

(*See e.g. Hayashi Fig. 4 and col. 1, lines 30-41, “When a transaction accesses the database for some data through the database management system, the data may already be in a buffer shared by transactions due to another transaction that previously accessed the same data, or the data must be transferred from the database to the shared buffer”*);

for a read-only transaction of a given database, creating a cache view of a given database using the given database’s transaction log (*Hayashi does not teach creating a cache view using the given database’s transaction log. However, Loaiza does, see col. 3, lines 12-28, “The database view of a recovery log (‘log view’) essentially provides a virtual database table that is constructed using data retrieved from one or more recovery logs... A SQL statement can be written to access or manipulate data in the virtual rows and columns of the log view” where the claimed “cache view” is the referenced “log view,” the claimed “transaction log” is the referenced “recovery log,” and the claimed “read-only transaction” is the referenced “SQL statement.” Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Loaiza’s teachings would have allowed Hayashi’s method and system to gain a virtual view of the database regardless of system*

crashes or other transactions, see Loaiza col. 3, lines 12-28), said cache view comprising particular database blocks of the shared cache that record a view of a particular version of the database at a given point in time (See e.g. Loaiza Table 1, col. 60, “Object ID Timestamp Block Addr.” and col. 3, lines 52-56, “To provide relational access to the records in this recovery log, a log view is defined having virtual columns for each item of information sought for each log record” which shows that the “log view” (i.e. the “cache view”) comprises database blocks, via the “Block Addr.,” that, using the “Timestamp,” record a view of the database at a particular point in time. Loaiza does not teach using database blocks from a shared cache. However, Hayashi does, see Fig. 4, “shared buffer 17,” referenced above. Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Hayashi’s teachings would have allowed Loaiza’s method and system to gain a common method of reusing database blocks, see Hayashi col. 1, lines 30-41);

creating a shadow cache for storing any database blocks that overflow said cache view during use of the cache view by the read-only transaction (See e.g. Hayashi Fig. 6 where, see col. 4, lines 8-18, “A buffer shared by the transactions is a bit map 30. The database 20 includes overflow pages 31. A database 20’ is used to nonvolatilize the contents of the shared buffer. The bit map 30 controls overflow pages 31” where the claimed “shadow cache” is the referenced “overflow pages 31”);

in conjunction with the cache view and the shadow cache, preserving a logical undo operation for the read-only transaction of the given database for logically undoing transactions which have begun but have yet to commit (Hayashi does not teach preserving a logical undo operation. However, Luo does, see Fig. 5A and col. 13, lines

Art Unit: 2168

16-21, “*a logical undo is performed.*” *Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Luo’s teachings would have allowed Hayashi’s method and system to gain the ability to rollback the user’s view of the database without effecting the database, see Luo col. 13, lines 16-36); and*

performing the logical undo operation in order to reconstruct a transactionally consistent prior version of the given database upon starting the read-only transaction, thereby returning a result comprising a transactionally consistent version of the given database supporting read-only uses (*Hayashi does not teach performing a logical undo operation. However, Luo does, see Fig. 5A and col. 13, lines 16-21, “In the logical undo, the database system looks for another tuple in the join view JV that has the attribute values (1, 2). That other tuple is changed to the value (1, 1) for a logical undo of transaction T₁, shown as 156 in FIG. 5A.” Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Luo’s teachings would have allowed Hayashi’s method and system to gain the ability to rollback the user’s view of the database without effecting the database, see Luo col. 13, lines 16-36).*

2. The method of claim 1, wherein during occurrence of the read-only transaction any database blocks associated with the cache view are not written from the shared cache to the given database (*See e.g. Hayashi Fig. 1 and col. 3, line 66 to col. 4, line 7, “The contents (i.e., pages) of the shared buffer are written back to the disk (i.e., data base 20) at a predetermined timing. Then, a log holding data updated by the last read or write operation will not be needed for recovering the contents of the buffer even if they are lost.*

The log buffer 16 temporarily stores pre-update and post-update logs of the shared buffers B1 and B2. The contents of the log buffer 16 are nonvolatilized by saving them in the log file 19 at a predetermined timing").

3. The method of claim 1, wherein the shadow cache is implemented via a temporary database table (*See e.g. Hayashi Fig. 2, "database 20" and "overflow pages 31").*

5. The method of claim 1, wherein the shadow cache is used only in the event the cache view overflows the cache view (*See e.g. Hayashi Fig. 2 where, see col. 4, lines 8-18, "A buffer shared by the transactions is a bit map 30. The database 20 includes overflow pages 31. A database 20' is used to nonvolatilize the contents of the shared buffer. The bit map 30 controls overflow pages 31").*

6. The method of claim 1, further comprising:

providing an allocation bitmap for indicating database blocks in use in the shadow cache (*See e.g. Hayashi Fig. 2, "bit map 30" and "overflow pages 31").*

7. The method of claim 6, further comprising:

upon completion of the read-only transaction, deleting the shadow cache by updating the allocation bitmap for allocated database blocks (*See e.g. Hayashi col. 4, lines 8-18, "any bit of the bit map 30 will be ON when a corresponding one of the overflow pages 31 is in use and OFF when the corresponding page is unused").*

8. The method of claim 1, wherein the shadow cache comprises a temporary database table including a first column for maintaining a block number of a cache view block having undo/redo records applied to it and a second column for maintaining a block number in a temporary database allocated to save off a modified block from the cache

view (See e.g. Hayashi col. 5, lines 30-46, "The database contains page data and a table showing relationships between page numbers and locations on the disk").

10. The method of claim 1, further comprising:

upon termination of the read-only transaction, marking the cache view as closed (See e.g. Hayashi Fig. 1 and col. 3, line 66 to col. 4, line 7, "The log buffer 16 temporarily stores pre-update and post-update logs of the shared buffers B1 and B2. The contents of the log buffer 16 are nonvolatilized by saving them in the log file 19 at a predetermined timing").

12. The method of claim 1, further comprising:

reusing the cache view created for the read-only transaction for other read-only transactions, which start within a specified period of time following the start of the read-only transaction (See e.g. Hayashi col. 3, lines 9-17, "A log buffer 16 stores pre-update and post-update logs" where no new log is created unless there is a write, thus the last update log is used for the next transaction).

13. The method of claim 1, further comprising:

detecting the read-only transaction (See e.g. Hayashi col. 1, lines 22-29, "Application programs running on a computer create transactions to query the database management system"); and

upon occurrence of write operations, adding back link log records to the database's transaction log that serve to link together blocks of the transaction log that pertain to the read-only transaction (See e.g. Hayashi col. 3, line 66 to col. 4, line 7, "The contents (i.e., pages) of the shared buffer are written back to the disk (i.e., data base 20) at a predetermined timing. Then, a log holding data updated by the last read or write

operation will not be needed for recovering the contents of the buffer even if they are lost" and Fig. 2, "log buffer 16" where the update logs are linked together).

15. A computer-readable medium having processor-executable instructions for performing the method of claim 1 (See e.g. Hayashi col. 1, lines 61-63, "The present invention relates to a method of and an apparatus for" where an "apparatus" implies "A computer-readable medium having processor-executable instructions").

16. A downloadable set of processor-executable instructions for performing the method of claim 1 (See e.g. Hayashi col. 1, lines 61-63, "The present invention relates to a method of and an apparatus for" where an "apparatus" implies "A downloadable set of processor-executable instructions").

17. A database system capable of restoring databases to a consistent version, the system comprising:

a log manager module which manages a transaction log of the database system (See e.g. Hayashi col. 2, lines 20-27, "a log buffer for temporarily storing pre-update and post-update logs, a log file for storing the pre-update and post-update logs");

a cache manager module for managing a shared cache that stores database blocks for use by multiple databases (See e.g. Hayashi Fig. 4 and col. 1, lines 30-41, "When a transaction accesses the database for some data through the database management system, the data may already be in a buffer shared by transactions due to another transaction that previously accessed the same data, or the data must be transferred from the database to the shared buffer") and creating a cache view of a given database created using the transaction log of the given database, said cache view being created in response to a read-only transaction of the given database (Hayashi does not teach creating a cache

view using the given database's transaction log. However, Loaiza does, see col. 3, lines 12-28, "The database view of a recovery log ('log view') essentially provides a virtual database table that is constructed using data retrieved from one or more recovery logs... A SQL statement can be written to access or manipulate data in the virtual rows and columns of the log view" where the claimed "cache view" is the referenced "log view," the claimed "transaction log" is the referenced "recovery log," and the claimed "read-only transaction" is the referenced "SQL statement." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Loaiza's teachings would have allowed Hayashi's method and system to gain a virtual view of the database regardless of system crashes or other transactions, see Loaiza col. 3, lines 12-28), said cache view comprising particular database blocks of the shared cache that record a view of a particular version of the database at a given point in time (See e.g. Loaiza Table 1, col. 60, "Object ID Timestamp Block Addr." and col. 3, lines 52-56, "To provide relational access to the records in this recovery log, a log view is defined having virtual columns for each item of information sought for each log record" which shows that the "log view" (i.e. the "cache view") comprises database blocks, via the "Block Addr.," that, using the "Timestamp," record a view of the database at a particular point in time. Loaiza does not teach using database blocks from a shared cache. However, Hayashi does, see Fig. 4, "shared buffer 17," referenced above. Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Hayashi's teachings would have allowed Loaiza's method and system to gain a common method of reusing database blocks, see Hayashi col. 1, lines 30-41); wherein the

cache manager utilizes a shadow cache for storing any database blocks that overflow said

cache view during use of the cache view by the read-only transaction (See e.g. *Hayashi*

Fig. 6 where, see col. 4, lines 8-18, "A buffer shared by the transactions is a bit map 30.

The database 20 includes overflow pages 31. A database 20' is used to nonvolatilize the

contents of the shared buffer. The bit map 30 controls overflow pages 31" where the

claimed "shadow cache" is the referenced "overflow pages 31"); and

a transaction manager module for performing read-only transactions of the database system and which performs a logical undo operation for the read-only transaction of the given database for logically undoing transactions which have begun but have yet to commit in order to reconstruct a transactionally consistent prior version of the given database upon starting the read-only transaction, thereby returning a result comprising a transactionally consistent version of the given database supporting read-only uses (*Hayashi does not teach performing a logical undo operation. However, Luo does, see Fig. 5A and col. 13, lines 16-21, "In the logical undo, the database system looks for another tuple in the join view JV that has the attribute values (1, 2). That other tuple is changed to the value (1, 1) for a logical undo of transaction T₁, shown as 156 in FIG. 5A."* Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Luo's teachings would have allowed Hayashi's method and system to gain the ability to rollback the user's view of the database without effecting the database, see Luo col. 13, lines 16-36).

18. The system of claim 17, wherein during occurrence of the read-only transaction any database blocks associated with the cache view are not written from the

shared cache to the given database (See e.g. Hayashi col. 3, line 66 to col. 4, line 7, "The contents (i.e., pages) of the shared buffer are written back to the disk (i.e., data base 20) at a predetermined timing. Then, a log holding data updated by the last read or write operation will not be needed for recovering the contents of the buffer even if they are lost. The log buffer 16 temporarily stores pre-update and post-update logs of the shared buffers B1 and B2. The contents of the log buffer 16 are nonvolatilized by saving them in the log file 19 at a predetermined timing").

19. The system of claim 17, wherein the shadow cache is implemented via a temporary database table (See e.g. Hayashi Fig. 2, "database 20" and "overflow pages 31").

21. The system of claim 17, wherein the shadow cache is used only in the event the cache view overflows the cache view (See e.g. Hayashi Fig. 2 where, see col. 4, lines 8-18, "A buffer shared by the transactions is a bit map 30. The database 20 includes overflow pages 31. A database 20' is used to nonvolatilize the contents of the shared buffer. The bit map 30 controls overflow pages 31").

22. The system of claim 17, wherein said cache manager maintains an allocation bitmap indicating database blocks in use in the shadow cache (See e.g. Hayashi Fig. 2, "bit map 30" and "overflow pages 31").

23. The system of claim 22, wherein said cache manager deletes the shadow cache by updating the allocation bitmap for allocated database blocks (See e.g. Hayashi col. 4, lines 8-18, "any bit of the bit map 30 will be ON when a corresponding one of the overflow pages 31 is in use and OFF when the corresponding page is unused" where making all bits OFF would effectively remove them from the "shared buffer").

24. The system of claim 17, wherein the shadow cache comprises a temporary database table including a first column for maintaining a block number of a cache view block having undo/redo records applied to it and a second column for maintaining a block number in a temporary database allocated to save off a modified block from the cache view (See e.g. Hayashi col. 5, lines 30-46, “The database contains page data and a table showing relationships between page numbers and locations on the disk”).

26. The system of claim 17, wherein said cache manager marks the cache view as closed, upon termination of the read-only transaction (See e.g. Hayashi col. 3, line 66 to col. 4, line 7, “The log buffer 16 temporarily stores pre-update and post-update logs of the shared buffers B1 and B2. The contents of the log buffer 16 are nonvolatilized by saving them in the log file 19 at a predetermined timing”).

28. The system of claim 17, wherein said cache manager reuses the cache view created for the read-only transaction for other read-only transactions which start within a specified period of time following the start of the read-only transaction (See e.g. Hayashi col. 3, lines 9-17, “A log buffer 16 stores pre-update and post-update logs” where no new log is created unless there is a write, thus the last update log is used for the next transaction).

29. The system of claim 17, wherein said log manager detects the read-only transaction, and adds back link log records to the transaction log that serve to link together blocks of the transaction log that pertain to the read-only transaction (See e.g. Hayashi col. 1, lines 22-29, “Application programs running on a computer create transactions to query the database management system” and col. 3, line 66 to col. 4, line 7, “The contents (i.e., pages) of the shared buffer are written back to the disk (i.e., data

base 20) at a predetermined timing. Then, a log holding data updated by the last read or write operation will not be needed for recovering the contents of the buffer even if they are lost" and Fig. 2, "log buffer 16" where the update logs are linked together).

Claims 11 and 27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hayashi et al., U.S. 5,715,447 (Hayashi), in view of *The Authoritative Dictionary of IEEE Standards Terms, Seventh Edition*, IEEE Press, 2000 (IEEE).

11. The method of claim 10, further comprising:

when new block allocations need to be made in the shared cache, traversing the shared cache looking for database blocks to purge; and

purging database blocks from any read-only view that have been marked as closed (*Hayashi does not teach purging database blocks. However, IEEE does, see "garbage collection (B) A database reorganization technique in which the contents of a database are made more compact by physically deleting garbage such as records that have been deleted logically but remain physically in the database" and "cache (2) A small portion of high-speed memory used for temporary storage of frequently-used data, instructions, or operands."* Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because IEEE's teachings would have allowed Hayashi's method and system to gain the ability to compact the contents of the database view, see "garbage collection (B) A database reorganization technique in which the contents of a database are made more compact by physically deleting garbage such as records that have been deleted logically but remain physically in the database").

27. The system of claim 26, wherein said cache manager traverses the shared cache looking for database blocks to purge, and purges database blocks from any read-only view that have been marked as closed when new block allocations need to be made in the shared cache (*Hayashi does not teach purging database blocks. However, IEEE does, see “garbage collection (B) A database reorganization technique in which the contents of a database are made more compact by physically deleting garbage such as records that have been deleted logically but remain physically in the database” and “cache (2) A small portion of high-speed memory used for temporary storage of frequently-used data, instructions, or operands.” Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because IEEE’s teachings would have allowed Hayashi’s method and system to gain the ability to compact the contents of the database view, see “garbage collection (B) A database reorganization technique in which the contents of a database are made more compact by physically deleting garbage such as records that have been deleted logically but remain physically in the database”*).

Claims 14 and 30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hayashi et al., U.S. 5,715,447 (Hayashi), in view of Raz, U.S. 5,701,480 (Raz).

14. The method of claim 13, further comprising:
if the read-only transaction must be undone, using the back link log records to skip portions of the transaction log that are irrelevant for undoing an uncommitted write transaction, wherein the back link log records are only generated in the transaction log when there are active read only transactions (*Hayashi does not teach skipping portions of*

the transaction log. However, Raz does, see col. 62, lines 3-17, "the computer 20 processes transactions using an 'undo' recovery mechanism that provides very fast recovery because only the effects of failed transactions must be undone." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Raz's teachings would have allowed Hayashi's method and system to gain greater efficiency by not undoing redundant or unnecessary transactions, see Raz col. 62, lines 3-17).

30. The system of claim 29, wherein said log manager uses the back link log records to skip portions of the transaction log that are irrelevant for undoing the write transaction, wherein the back link log records are only generated in the transaction log when there are active read only transactions (*Hayashi does not teach skipping portions of the transaction log. However, Raz does, see col. 62, lines 3-17, "the computer 20 processes transactions using an 'undo' recovery mechanism that provides very fast recovery because only the effects of failed transactions must be undone." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Raz's teachings would have allowed Hayashi's method and system to gain greater efficiency by not undoing redundant or unnecessary transactions, see Raz col. 62, lines 3-17).*

Response to Arguments

After further review and consideration, the Examiner has withdrawn the 35 U.S.C. 112 rejections.

Art Unit: 2168

As per Applicant's arguments that claims 1-3, 5-8, 10-19, 21-24, and 26-30 are statutory under 35 U.S.C. 101, the Examiner respectfully disagrees. The disclosed subject matter provides for creating a view of the database at a particular point in time for performing read-only transactions. This produced result remains in the abstract because there is no result sent to another system or output displayed to a user. Rather, the method is a manipulation of data performed within the database system then made available to a read-only transaction. Making something available is not a tangible result.

As per Applicant's argument that claims 17-19, 21-24, and 26-30 are statutory under 35 U.S.C. 101, the Examiner respectfully disagrees. Independent claim 17 recites a system (interpreted to be a physical machine of some kind) which includes a "log manager," "cache manager," and a "transaction manager." While these various managers could be embodied in hardware, there is nothing in the claims to indicate that they actually are. A proper system claim generally includes a processor and a memory containing instructions that when executed by the processor cause the method to be executed. Thus, a proper system claim requires real hardware elements that are interrelated.

As per Applicant's argument that Hayashi does not teach the claimed cache view, the Examiner agrees and has cited Loaiza as citing the necessary limitations. See particularly col. 3, lines 12-28, "The database view of a recovery log ('log view') essentially provides a virtual database table that is constructed using data retrieved from one or more recovery logs... A SQL statement can be written to access or manipulate data in the virtual rows and columns of the log view" where the claimed "cache view" is the referenced "log view," the claimed "transaction log" is the referenced "recovery log," and

the claimed “read-only transaction” is the referenced “SQL statement.” See further Table 1, col. 60, “Object ID Timestamp Block Addr.” and col. 3, lines 52-56, “To provide relational access to the records in this recovery log, a log view is defined having virtual columns for each item of information sought for each log record” which shows that the “log view” (i.e. the “cache view”) comprises database blocks, via the “Block Addr.,” that, using the “Timestamp,” record a view of the database at a particular point in time.

Loaiza does not teach using database blocks from a shared cache. However, Hayashi does, see Fig. 4, “shared buffer 17,” referenced above.

As per Applicant’s argument that Hayashi does not teach the claimed shadow cache, the Examiner respectfully disagrees. Applicant has not defined the shadow cache in claims 1 and 17, only a use is recited, which carries no patentable weight. Thus, the Examiner is free to apply the broadest reasonable interpretation to the limitation. While Hayashi discloses a cache of pages overflowing from the shared buffer, it could easily be used to store overflow pages from Loaiza’s log view.

As per Applicant’s argument that Hayashi does not disclose a logical undo operation, the Examiner agrees, and has cited Luo as teaching the necessary limitations. See particularly Fig. 5A and col. 13, lines 16-21, “In the logical undo, the database system looks for another tuple in the join view JV that has the attribute values (1, 2). That other tuple is changed to the value (1, 1) for a logical undo of transaction T_1 , shown as 156 in FIG. 5A.” Hayashi teaches a physical undo operation, so it would be obvious to use a logical undo instead in light of Luo’s disclosure.

Conclusion

Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Aaron Sanders whose telephone number is 571-270-1016. The Examiner can normally be reached on M-F 9:00a-4:00p.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Tim Vo can be reached on 571-272-3642. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/AJS/
Aaron J. Sanders
Examiner
26 November 2007

CHC
12/4



CHRISTIAN CHACE
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100